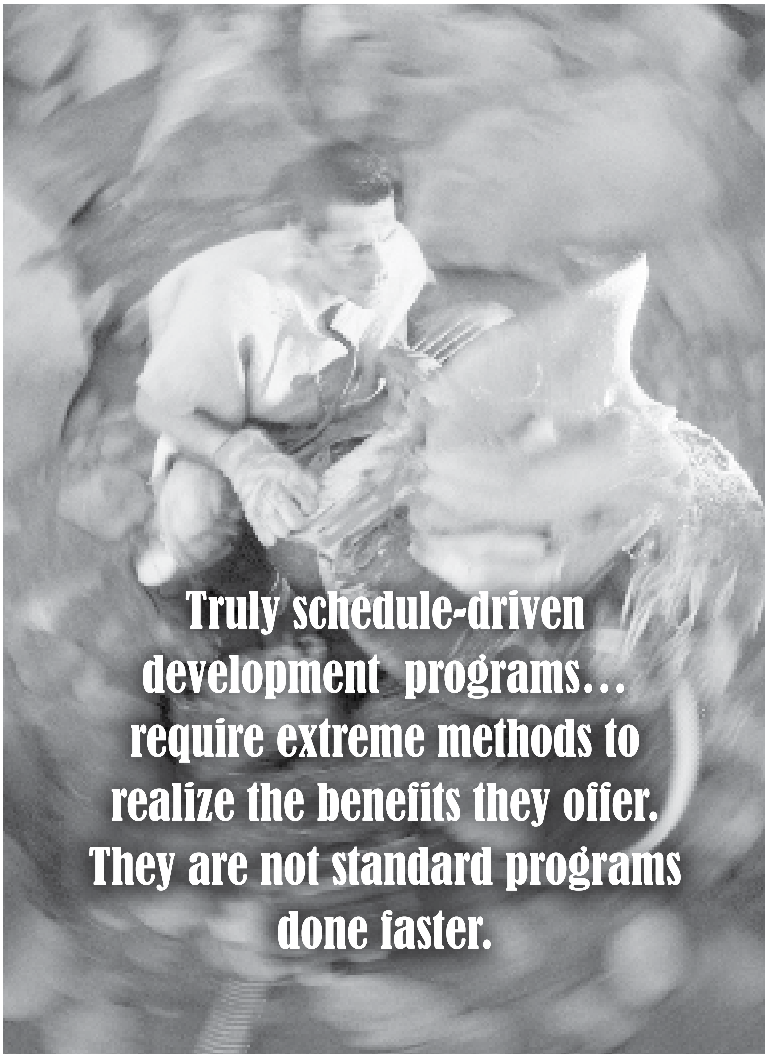
Survival Guide for Truly Schedule-Driven Development Programs

Sue Neves ▪ Jack Strauss

Schedule-driven development programs are different from standard acquisition efforts. All programs have a measure of schedule pressure. Once baselined, the "iron triangle" of cost, schedule, and technical scope is at play. But truly schedule-driven development programs behave differently and have different needs. Attempting to plan, execute, and manage a truly schedule-driven development effort as if it were a standard acquisition program done faster will not work, will slip, will cost more—and will probably get you fired.



Truly schedule-driven development programs… require extreme methods to realize the benefits they offer. They are not standard programs done faster.

For standard acquisition programs, the delivery of capability/maturity, in terms of program structure and tasks, is well known and fits nicely into the Department of Defense methods, processes, and culture. This is shown by the solid line in Figure 1. A schedule-driven development effort has different behavior. It surges, is less predictable, and does not fit as well into the DoD methods of oversight and reporting. Then why do it? The promise of the schedule-driven effort is that the capability can be delivered before that same capability could be delivered through the standard approach, as shown by the dotted line in the figure. The benefit is time savings (which may mean some cost savings) or a critical combat capability delivered when promised or earlier, or both.

If, however, the schedule-driven program is not resourced correctly in the early phases of the effort, it will slip. The DoD acquisition system, which has been stressed by the very existence of the effort in the first place, is now required to fix what looks like very poor program performance when compare the expectations of a standard program, as depicted in Figure 2.

Our experience shows that the factors we are going to discuss are key to determining the ability to actually accomplish a truly schedule-driven development program. Clearly there are other factors, but the ones we found were the most obvious, at least in hind-sight. Use these factors to plan a program for success if you are in the planning phase, or use them to diagnose an ongoing effort.

Lean Requirements

At the very onset of the system design and development portion of the program, all trade space in program requirements should be reviewed and identified. Getting part way through the program, then discovering the contractor is

Neves is an electronics engineer with 24 years of procurement engineering experience. She has bachelor's and master's degrees in systems engineering from Wright State University. Strauss, a systems engineer with 30 years of experience, has a master's degree in electrical engineering from the Air Force Institute of Technology and a postgraduate certificate from Stanford University.

Figure 1. **Schedule-Driven versus Standard Program—Well Executed**

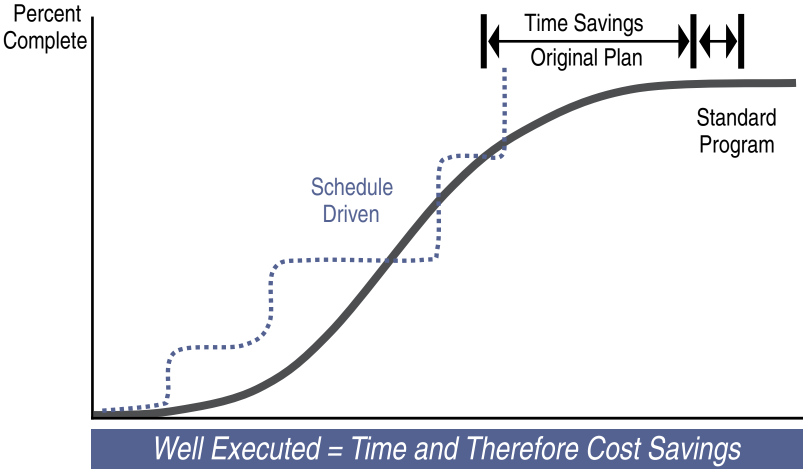
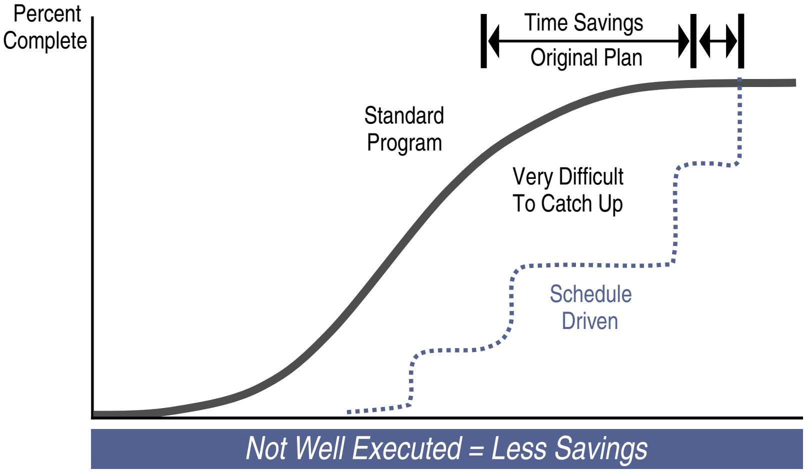


Figure 2. **Schedule-Driven versus Standard Program—Poorly Executed**



struggling to comply with a tradable feature or capability, and then bargaining away that trade space is wasteful of resources and precious time. Make the performance requirements as lean as possible right from the start; you don't have the luxury of time to massage the objective performance thresholds. All requirements should be at the key performance parameter threshold level, with objective and threshold being equal in every case. The rationale is this: Pass/fail thresholds are much easier and clearer to meet, defend, and communicate. This will enable you, as the government procuring official, to stand firm while insisting the lean requirements be met.

Development Capacity

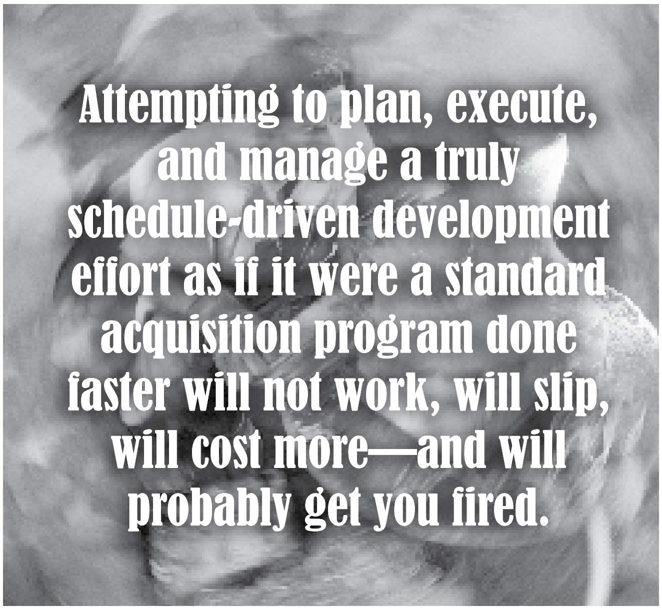
Development capacity is defined as the actual capacity to fabricate your development products. Your capacity must be at least twice the nominal requirement. For example, if you are going to fabricate 10 systems over a two-year period, then your capacity must be planned for a rate that would yield 20 over that period. Since your program is still developing the system while testing it and starting to produce it, many—at least half—of the development assets will require updates as the design matures. The only way to facilitate the updates is to have the excess (with respect to nominal) capacity to accommodate them. Please note that the recommendation to double capacity is conservative; quadrupling would be better. Optimization in this area is for standard programs and production efforts, not for truly schedule-driven development efforts. If you optimize too early, you doom your program to being unrecoverable in schedule if testing reveals the need to change (and that's a certainty in a development program). Also, be sure that the doubled capacity comes on line no later than midway through the program. If it is any later than that, you discount its impact and ability to recover. Be creative with leases or loans or procurements of equipment, but make sure it is there when you need it—all of it—for as long as you need it. Your capacity will be your last line of defense when your design is maturing. Expertise in this area allows you to reuse most, if not all, of this capacity in your production phases and thus control program costs downstream.

Development Asset Procurement

Procure 20 percent more development assets than nominal requirements. If you think you need 10 prototype systems or engineering development models, then procure 12. You will, in fact, drop, overheat, or just wear out your engineering development models. Additionally, you must have enough assets to accomplish simultaneous test and lab/support activities. If you don't have enough assets to replicate flight test in the exact, identical configuration in your labs, your program will slip as you attempt to complete development on the flight test asset, which is ill-suited for the task and extremely costly.

Consistent Engineering Discipline

Insist on engineering discipline. Cutting corners here is exactly the wrong thing to do. The only sure way to make decisions fast and make them only once, is to have all the data and to follow disciplined engineering methods. These data include root cause analysis, test results, results from modeling and simulation, and the outputs from proven engineering methods. Disciplined configuration management is a real key here. It is critical to understand exactly what is being changed and why. To paraphrase Sir Arthur Conan Doyle, "Never guess, as it is a mistake to theorize before one has data because one begins to twist facts to suit theories, instead of theories to suit facts."



Attempting to plan, execute, and manage a truly schedule-driven development effort as if it were a standard acquisition program done faster will not work, will slip, will cost more—and will probably get you fired.

Award Fee

Be very careful with award fees. Incentivizing contractors by establishing objective award fee criteria to provide a capability by a certain date has been proven to affect their behavior in unintended ways. Technical and cost discipline gets compromised to favor the schedule-driven objective event. For example, we have witnessed proposed specification changes to allow for delivery of non-compliant assets, not because the specification change was warranted or technically defendable, but to meet an objective award fee date. The real trick is to paper the deal with clear definitions of performance thresholds and system configuration for the capability. Additionally and equally important is a clear means of government acceptance of the capability (for example, the DD 250 material inspection and receiving report, which is the government's method for accepting delivery of systems). However, don't underestimate the level of amateur lawyering in which your contractor will engage to campaign for the objective award fee, for political reasons, when the objective was clearly not met. Under extreme schedule and award fee pressure, malicious compliance may emerge (and in our experience has) for any unclear definition, configuration, criterion, or acceptance method. Negotiation tactics come into play as people try to argue that the award fee words did not really mean what they clearly said. Beware of late-game arguments that start with the words "its intended purpose…." It is our strongest recommendation that only subjective criteria be applied to critical schedule-driven program events. That enables the government procurement team to exercise that subjectivity with awarding the fee, as we've seen the inclination to do with objective criteria, without losing credibility by arguing semantics and thus compromising its integrity by contradicting its own award fee plan. If the fee-determining official is provided clear and unambiguous subjective award fee criteria matched to real program status, you have done your job, and the subjective criteria can be objectively applied to your program. If this line is held for two consecutive award fee periods, all participants will trust the process, and the tool becomes powerful rather than an extraordinary distraction.

Approval Authority of Products and Documents

The flip side of what we just said is that the government must not trade away approval authority in the interest of saving time. The government program offices must be resourced so they do not fall into the trap of streamlining to the point of waiving approval for acceptance test procedures, qualification procedures, specifications for critical subassemblies, producibility and manufacturing plans, logistics support plans, and so on. Without government approval of key acceptance criteria, the government may find itself contractually bound to accept a non-performing capability and paying an award fee on top of it. (This is also known as accepting garbage on time.)

Funding Risk Areas

Generously fund the technical risk area, and don't be afraid to use it. Push your contractor—and yourself—to actually develop the risks and their mitigation plans. A few extra days at early program management reviews and design reviews are a small price to pay for this contingency. Risk plans that merely exist in presentation material and have not been developed so that schedule, performance, and cost impacts are known in terms of the program integrated master schedule, system specification, test plans, and development capacity are worse than having no risk management at all. Your leadership will think risk plans exist when they really don't. Or, equally as bad, priced risks will show up in your cost estimates for the next phase as a factored increase, and you will have no technical rationale to support otherwise.

Truly schedule-driven development programs are rare. They require extreme methods to realize the benefits they offer. They are not standard programs done faster. If you can't afford to implement the measures discussed above, then don't start. If you find yourself in a truly schedule-driven development program that has not been adequately resourced, then consider the steps outlined above. Influence change in those areas anywhere you can; some can be modified, even if the program is already under way. By doing so, you may be able to reduce the risks of a schedule-driven program and minimize the impact when the going gets tough and the pressure against the program schedule increases.

The authors welcome comments and questions and can be contacted at susan.neves@wpafb.af.mil and jstrauss@xcelsi.com.